# Fault Diagnosis and Prognosis Based on Lebesgue Sampling

Bin Zhang and Xiaofeng Wang

*Department of Electrical Engineering, University of South Carolina, Columbia SC, USA*
*zhangbin@cec.sc.edu*
*wangxi@cec.sc.edu*

**ABSTRACT**

Traditional fault diagnosis and prognosis (FDP) approaches are based on periodic sampling, *i.e.* samples are taken and algorithms are executed both in a periodic manner. As the volume of sensor data and complexity of algorithms keep increasing, the bottleneck of FDP is mainly the limited computational resources, which is especially true for distributed applications where FDP functions are deployed on microcontrollers and embedded systems with limited computation resources. This paper introduces the concept of Lebesgue sampling in FDP and proposes a Lebesgue sampling based fault diagnosis and prognosis (LS-FDP) framework. In the proposed LS-FDP, a novel diagnostic philosophy of "execution only when necessary" is developed in computation cost reduction and uncertainty management. For prognosis, different from traditional approaches in which the prognostic horizon is on the time axis, the proposed approach defines prognostic horizon on the state axis. With a reduced prognostic horizon, the LS-FDP naturally benefits the uncertainty management. The goal is to create the fundamental knowledge for LS-FDP solutions that are cost-efficient, capable for the deployment on systems with limited computation sources, and supportive to the trend of distributed FDP schemes in complex systems. The design and implementation of LS-FDP based on particle filtering algorithms are presented with experimental results to verify the effectiveness of the proposed approaches.

## 1. INTRODUCTION

Integrated System Health Management is a critical capability required for many safety critical systems such as unmanned air/ground/sea vehicles, aircraft, power generation, nuclear power plants, and various industrial systems (Tang, Zhang, DeCastro, & Hettler, 2011; Tang, Hettler, Zhang, & DeCastro, 2011; DeCastro, Tang, & Zhang, 2011; Zhang, Tang, DeCastro, & Goebel, 2011; Balanban & Slonso, 2013). The fundamental enabling technologies of integrated system health management include sensing, data acquisition, fault diagnosis and prognosis (FDP), and decision-making, etc. Diagnosis and prognosis, as fundamental enabling techniques, are not new concepts (Tumer & Bajwa, 2004; Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2006; Zhang, Khawaja, Patrick, & Vachtsevanos, 2008; Schwabacher & Goebel, 2007). Diagnosis aims to monitor the health state of the component or the system such that the current health state can be obtained in real-time. The challenge in diagnosis is to detect potential faults as early and accurate as possible during the operation of a monitored system. Usually a fault cannot be measured directly. In Bayes theory, the fault state can be obtained by applying Bayesian estimation with a fault diagnostic model and a real-time measurement (Boskoski & Urevc, 2011; Zhang, Khawaja, Patrick, & Vachtsevanos, 2010; Zhang, Sconyers, et al., 2009; Zhang, Khawaja, et al., 2009; Li, Kurfess, & Liang, 2000; Goebel, Eklund, Hu, Avasarala, & Celaya, 2006; Goebel, Saha, & Saxena, 2008). In the context of fault diagnosis, the real-time measurements are often features or fault condition indicators extracted from raw measurements, such as vibration, current, voltage.

Prognosis refers to the generation of long-term predictions that describe the evolution of a fault and the estimation of the remaining useful life (RUL) of a failing component or subsystem. In reliability study, there are many diagnostic and prognostic approaches, such as Weibull-based risk distributions (Kaminskiy, 2005), the graphical reliability degradation modeling approach (Huang & Dietrich, 2005), and the degradation path curve approach (Lawless, 2003; Finkelstein, 2004; Yang, 2005), to name a few. For online prognosis, filter-based approaches are more promising, such as Kalman filter (Celaya, Saxena, & Goebel, 2012), extended Kalman filter (Saha, Goebel, Poll, & Christophersen, 2009), unscented Kalman filter (Anger, Schrader, & Klingauf, 2012), and particle filter (Zhang et al., 2010). Compared with many successful cases of diagnosis (Isermann, 2005; Zhong, Fang, & Ye, 2007; Hess & Wells, 2003; Zhang et al., 2010; Zhang, Sconyers, et al., 2009; Zhang, Khawaja, et al., 2009; Zhang et al., 2008; Oppenheimer & Loparo, 2002; Agogino, Bonis-

sone, Goebel, & Vachtsevanos, 2001; Jardine, Lin, & Banjevic, 2006), prognosis is more challenging (Schwabacher & Goebel, 2007; Vachtsevanos et al., 2006; Edwards, Orchard, Tang, Goebel, & Vachtsevanos, 2010; Usynin & Hines, 2007; Celaya et al., 2012). Major contributors to this difficulty include nonlinear nature of fault growth, absence of measurement, hybrid nature of fault modes, and various uncertainties.

A comparison of several prognostic approaches can be found in (Goebel et al., 2008). To evaluate the performance of FDP, different performance indexes were also developed (Saxena, Celaya, Saha, Saha, & Goebel, 2009; Orchard & Vachtsevanos, 2009). For diagnosis, the matrices are often related to false alarm rate, probability of detection, etc. For prognosis, most matrices are evaluated in terms of accuracy and precision of RUL estimation. These metrics are often offline evaluated when failure has been physically reached and is compared with the RUL estimation from prognosis.

Traditional ways to design FDP algorithms adopt periodic sampling (also called "Riemann sampling (RS)") where samples are taken in a periodic manner and the diagnostic and prognostic algorithms are executed at the same rate. A nice feature of FDP with this fixed time interval sampling is the easiness in analysis and design. However, it may be undesirable in many situations, from the computation-efficiency point of view. On the one hand, since the sampling period is determined according to the worst-case operating scenario, the FDP algorithm might be executed even if there is little new information actually present in the measurements. In other words, the algorithm may take greater utilization than it actually needs. This will result in significant overprovisioning of the real-time system hardware. On the other hand, when the fault grows very fast, it is expected to assign more resources to the FDP algorithm so that it can takes more frequent actions to provide accurate fault information, which obviously cannot be met by periodic sampling. For prognosis, RS-based FDP usually has a large prediction horizon, from the time that a fault is detected at very early stage to a future time instant that the fault grows to the failure threshold. This long-term prediction not only requires a lot computation resources, but also causes accumulation of uncertainties. The LS-FDP considers the prediction horizon in the fault dimension axis and described by the number fo fault states. This provides a straightforward means to conduct prognosis that requires little computation resources.

As the applications of FDP has increased rapidly, the heavy demand on computational resources makes existing FDP algorithms very hard to be deployed on embedded systems that are widely used but have very limited computation capabilities. This becomes the bottleneck that prevents the distribution of FDP algorithms in complex systems. To break this bottleneck, cost-efficient FDP solutions must be developed. With this vision, we propose the Lebesgue sampling-based FDP (LS-FDP) method, which is a cost efficient FDP approach where computation can be executed on an "as-needed" basis and is promising in reducing the computational cost compared with the traditional Riemann sampling-based FDP (RS-FDP) algorithms. In this new approach of FDP, the novelty comes from the concept of "Lebesgue sampling (LS)" (or "event-based sampling"). Contrast to conventional periodic sampling-based approaches, the computation in LS-FDP will be triggered only when an event takes place, and the prognosis will be executed based on the LS-based model whose states are predefined according to the quantization level. With the feature of "execution only when necessary" in LS, the computation efforts in LS-FDP can be significantly reduced by eliminating unnecessary computation when fault growth is slow.

The paper is organized as follows: Section 2 provides an overview of the proposed LS-FDP framework. Section 3 develops a particle filtering based LS-FDP approach, which is followed by experimental results on an epicycle planetary gear box presented in Section 4. Section 5 gives the concluding remarks with some future research topics.

## 2. THE PROPOSED LS-FDP FRAMEWORK

This section will establish the complete LS-FDP framework with an overview of the proposed solutions. The unique innovative feature of the proposed LS-FDP is that the diagnostic and prognostic algorithm is no longer carried out in a fixed time interval. Instead, the diagnosis is carried out only when new measurements justify that the fault conditions have changes to warrant the execution. The LS-FDP framework is illustrated in Figure 1, which integrates external inputs, Lebesgue samples of feature and fault dimension, models for diagnosis and prognosis, and diagnostic and prognostic algorithms.

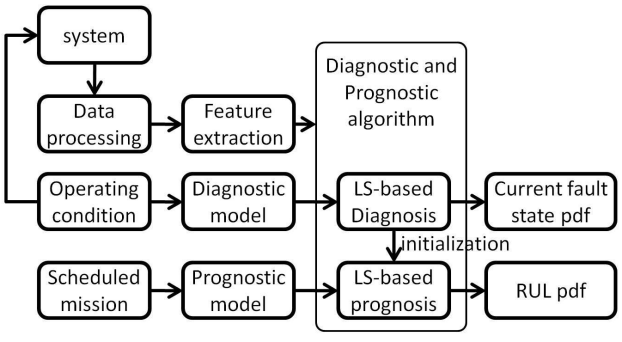



Figure 1. The implementation framework of LS-FDP

In this paper, our focus is the introduction of Lebesgue sampling into diagnosis and prognosis. Therefore, we will not discuss data collection, preprocessing, and feature extraction. After a feature has been successfully extracted from data to

indicate the growth of a fault, the performance and efficiency of FDP relies greatly on the dynamic model that describes the fault behavior, and the diagnostic and prognostic algorithms, which will be elaborated in the following sections.

### 2.1. Fault Mechanism Modeling

Assume that the actual fault growth dynamics can be described by the following continuous-time differential equation:

$$\dot{a} = F(a, u) \tag{1}$$

where $a$ is the fault dimension, $u$ is system input including items (such as external environmental factors and operating modes) that have impacts on fault growth, and $F(\cdot)$ is a nonlinear function that describes the fault growth under the current fault dimension with input $u$. The feature or condition indicator, denoted by $y$, is extracted from raw measurements and serves as the real-time measurement for FDP algorithm. Note that the mapping between $y$ and $a$ can be described by a nonlinear function $y = h(a)$. In most cases, $a$ is not measurable and $y = a$ is employed such that we can use $y$ to indicate fault $a$ directly. To simplify the description, we take $y = a$ in the following discussion.

To use this model in LS-FDP, we quantify the fault measurements. Lebesgue sampling basically takes samples when the difference between the current state and the last sampled state exceeds the pre-defined Lebesgue state length. Then the LS-based model of the fault dynamics in discrete-time can be described as follows:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t(D, \dot{\hat{a}}(t_k)) \tag{2}$$

where $\hat{a}(t_k)$ is the Lebesgue state, $t_k$ is the $k$th sampling instant, $D$ is the Lebesgue length, and $f_t(\cdot)$ is a nonlinear function.

In traditional prognostic algorithm, there are two steps of prognosis. The first step is the generation of a long-term predication for the fault state pdf estimation. This is obtained by recursive execution of the fault growth model. The second step is the estimation of RUL, which is essentially related to the probability of failure at future time instants. The RUL pdf is obtained by defining a failure threshold established from historical data or empirical knowledge and comparing this threshold with the long-term prediction of fault state at all the future time instants. Compared to diagnosis, prognosis requires much more computational resources mainly because of long-term predication, especially when the prediction horizon is large, which is not a rare case in FDP applications. To reduce computation time and resources, a new model is developed in the LS-FDP as follows:

$$t_{k+1} = t_k + g_t(D, \dot{\hat{a}}(t_k)) \tag{3}$$

Note that $\dot{\hat{a}}(t_k) = f(\hat{a}(t_k), u(t_k))$ and $g_t(D, \dot{\hat{a}}(t_k))$ is a nonlinear function. Rather than conducting a long-term prediction on the time axis, this model calculates the RUL on each Lebesgue state directly so that the prediction horizon is the number of Lebesgue states on the fault dimension axis. Since the number of Lebesgue states on the fault dimension axis is small, the prediction horizon for LS-based prognosis is small and will significantly reduce the computation.

### 2.2. The Concept of Lebesgue Sampling

The concept of Lebesgue sampling can be illustrated through an example of a crack on a planetary gear carrier plate in a helicopter main power transmission system (Zhang et al., 2010). The seeded crack starts to grow from an initial value of 1.34 inches to 7.67 inches in 1000 cycles of operation and the ground truth crack dimension growth is shown in Figure 2. It is clear that the fault growth in the range $R_1 = [50, 650]$ cycle is slower than that in the range $R_2 = [650, 750]$ cycle. Using Riemann sampling-based FDP with fix time interval, as shown in Figure 2(a), the FDP algorithms are executed at each cycle no matter if it is necessary. Since the fix time interval is selected according to the worst-case scenario to guarantee tracking accuracy for fault growth in range $R_2$, there are many unnecessary calculations in range $R_1$.

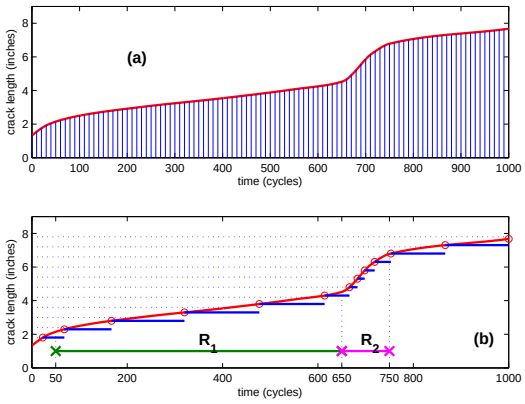



Figure 2. Illustration of LS. (a) RS with fixed time interval; (b) LS with fixed Lebesgue state length

Ideally, we expect to reduce the number of FDP execution in the range $R_1$ where the fault growth is slow so that more resources can be assigned to other tasks. In the range of $R_2$ where the fault growth becomes fast, we increase the number of FDP execution by assigning more resources to FDP tasks. This setting is desirable in FPGA-based embedded systems where resources are dynamically reconfigurable and are assigned to different tasks in realtime. With this configuration, a balance between computation and performance can be achieved. This strategy however involves time-varying sampling periods that is not an easy task within the Riemann sampling framework. With Lebesgue sampling, the realization of

this strategy becomes natural. By defining Lebesgue states on the vertical axis of fault dimension (crack length in this figure), fewer transitions between states are made when the fault growth is slow while more transitions are made when the fault growth is fast. For the example shown in Figure 2.(b), only 5 Lebesgue states are visited during the 550 cycles in $R_1$ and 4 states during the 100 cycles in $R_2$, which means that the FDP only needs to be executed 5 times during $R_1$ and 4 times during $R_2$. With this consideration, during $R_1$, more computation resources can be assigned to other tasks while only a little resources are needed for FDP. During $R_2$, more resources are assigned to FDP tasks so that the fault dimension can be tracked accurately.

### 2.3. Lebesgue Sampling-Based Diagnosis

In the LS-FDP framework, the range of the state $a(t)$ is partitioned into Lebesgue states $\{F_1, F_2, \cdots, F_f\}$, with which the diagnostic model is discretized. The diagnostic algorithm is executed when an event happens, *i.e.* the state $a(t)$ changes from one Lebesgue state to another one (McCann & Le, 2008; Astrom & Bernhardsson, 1999). The time instant when an event is generated is called the "event stamp". The sequence of the event stamps is denoted as $t_1, t_2, t_3, \cdots$, which formulates a time series that can be used as the input of run-time diagnostic algorithms such as a Kalman filter-based or particle filter-based algorithm (Morales-Menendez, de Freitas, Monterrey, Freitas, & Poole, 2002; de Freitas, 2002; Zhang et al., 2010; Zhang, Sconyers, et al., 2009; Orchard, Hevia-Koch, Zhang, & Tang, 2013). The output of diagnostic algorithm is the current fault state distribution at these event stamps and the probability of fault detection. The implementation procedure of the Lebesgue sampling-based diagnosis can be illustrated in the flow charts shown in Figure 3.

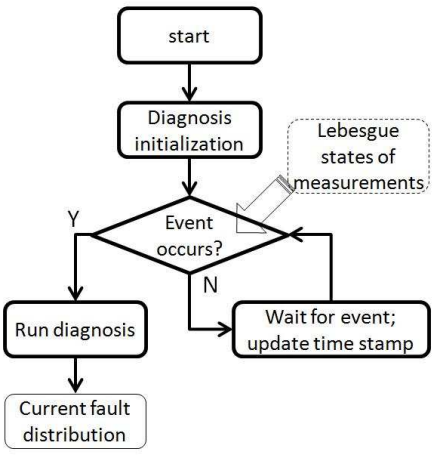



Figure 3. Flow chart of Lebesgue sampling-based diagnosis

### 2.4. Lebesgue Sampling-Based Prognosis

When a fault is detected at $t_d$, a time distribution is initialized as the initial condition for prognosis. By Riemann sampling-based prognosis, the prediction is conducted from the current time instant $t_{current}$ to future time instants till $t_{fail}$ when the fault state reaches a failure threshold $F_f$. The prognostic horizon $[t_{current}, t_{fail}]$ is usually large, especially at the early stage of the fault or when the fault growth is slow. The prediction calculates the fault state at each fixed time interval, which is demanding on the computational resources. Moreover, prognostic uncertainty will grow rapidly with large prediction horizon.

With LS, a new prognostic philosophy is proposed. Suppose that the fault is detected at Lebesgue state $F_d$, then we consider the discretized prognostic model with Lebesgue states $\{F_d, F_{d+1}, \cdots, F_f\}$. The prognostic algorithm is implemented, together with the LS-based prognostic model, to calculate the distributions of operation time when the fault state reaches different Lebesgue states $\{F_d, F_{d+1}, \cdots, F_f\}$. Meanwhile, it will provide a RUL estimation on Lebesgue state $F_f$. Note that the prognostic horizon can be controlled by adjusting Lebesgue state length. Increasing the Lebesgue state length will decrease the number of events, which will reduce the required computational resources. The implementation procedure of the Lebesgue sampling-based prognosis can be illustrated in the flow charts shown in Figure 4.

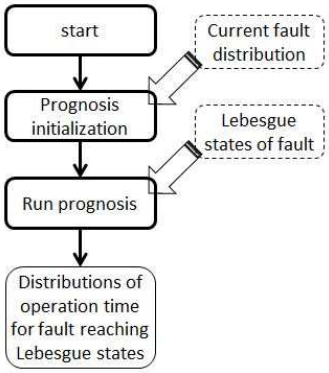



Figure 4. Flow chart of Lebesgue sampling-based prognosis

## 3. METHODOLOGY DEVELOPMENT

### 3.1. Particle Filter for LS-Based Diagnosis

The fault diagnosis is basically a state estimation problem, which can be handled in a Bayesian framework. Mathematically, assume the unobserved fault process $X$ to be a Markov process characterized by initial distribution $p(x_0)$ and the transition probability $p(x_k|x_{k-1})$ defined by $x_k = f_k(x_{k-1}, \omega_k)$ with $\omega_k$ being the process noise. The subscript $k$ represents the $k$th event stamp caused by the transition of Lebesgue

states. The observations $Y$ are assumed to be conditionally independent given $X$. The distribution of $(Y_k|X_k)$ is defined by $y_k = h_k(x_k, v_k)$ with $v_k$ being observation noise. Let $x_{0:k} = \{x_0, \cdots, x_k\}$ and $y_{1:k} = \{y_1, \cdots, y_k\}$ denote the state and the observation up to the $k$th event. It is of interest to estimate the *posterior distribution* $p(x_{0:k}|y_{1:k})$. The task can be achieved by two sequential steps, prediction and filtering.

In most nonlinear cases, however, analytical solutions do not exist. Alternatively, sequential Monte Carlo (SMC) methods, such as particle filter (Zhang et al., 2010), provide approximate solution to state estimation that is used for fault diagnosis.

Assume that a set of $N$ particles $(w_{k-1}^{(i)}, x_{0:k-1}^{(i)})$ is available such that they can be used to approximate a desired distribution $\pi_{k-1}(x_{0:k-1})$, where the superscript $i = 1, 2, \cdots, N$ denotes $N$ particles located at $x_{0:k-1}^{(i)}$ and $w_{k-1}^{(i)}$ is the weight of the $i$th particle at the $(k-1)$th event. The objective is to efficiently obtain a new set of $N$ particles $(w_k^{(i)}, \bar{x}_{0:k}^{(i)})$ that can approximate the distribution $\pi_k(x_{0:k})$, where $\bar{x}_{0:k}^{(i)}$ denotes location of $N$ new particles. In the context of SMC methodology, a Monte Carlo approximation can be obtained as:

$$\pi_k(x_{0:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta\left(x_{0:k} - \bar{x}_{0:k}^{(i)}\right). \tag{4}$$

with $\sum_{i=1}^{N} w_k^{(i)} = 1$, where $\delta$ denotes the Dirac-delta function. The weight can be updated in a recursive formula as:

$$\begin{aligned} w\left(\bar{x}_{0:k}^{(i)}\right) &= w_{k-1}^{(i)} h_k\left(y_{1:k}|\bar{x}_{0:k}^{(i)}\right) \quad \text{and} \\ w_k^{(i)} &= \frac{w\left(\bar{x}_{0:k}^{(i)}\right)}{\sum_{i=1}^{N} w\left(\bar{x}_{0:k}^{(i)}\right)}. \end{aligned} \tag{5}$$

To implement the above mentioned particle filtering based fault diagnosis with LS, an LS-based diagnostic model is given by:

$$\begin{cases} \begin{bmatrix} x_{d,1}(t_{k+1}) \\ x_{d,2}(t_{k+1}) \end{bmatrix} = f_b\left(\begin{bmatrix} x_{d,1}(t_k) \\ x_{d,2}(t_k) \end{bmatrix} + n(t_k)\right) \\ \hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t\left(D, \dot{\hat{a}}(t_k)\right) \cdot x_{d,2}(t_k) + \omega_a(t_k) \\ y(t_k) = \hat{a}(t_k) + v(t_k) \end{cases} \tag{6}$$

with nonlinear mapping $f_b(x)$ is given by

$$f_b(x) = \begin{cases} [1 \quad 0]^T, & \text{if } \|x - [1 \quad 0]^T\| \leq \|x - [0 \quad 1]^T\| \\ [0 \quad 1]^T, & \text{otherwise.} \end{cases}$$

and the initial condition is given by:

$$\begin{bmatrix} x_{d,1}(0) \\ x_{d,2}(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

where $x_{d,1}$ and $x_{d,2}$ are a collection of Boolean states that indicate *normal* and *faulty* conditions, respectively, $\hat{a}$ is the Lebesgue state that represents the fault dimension, $\omega_a$ and $v$ are process and observation noises, respectively, $n$ is independent and identically distributed uniform white noise, and $u$ is the external input. In this equation, $t_k$ is the event stamp indicating that there is a state transition event. As assumed earlier, the feature value $y(t_k)$ indicates the fault value $\hat{a}(t_k)$ directly, in order to simplify the description.

During the process of LS-based diagnosis, the diagnostic algorithm is executed only when the new measurement $y$ shows that significant information is included. For this purpose, the range of feature (also fault in this case) is divided into a series of Lebesgue states. If two successive measurements cause a transition of Lebesgue state, the diagnostic algorithm will be executed. Otherwise, it won't be executed.

### 3.2. Particle Filter for LS-Based Prognosis

Prognosis estimates the RUL. In traditional RS-based prognosis, the prediction is carried out with fix time interval from the current time instant $t_{current}$ to the time instant $t_{fail}$ that fault state reaches failure threshold $F_f$. The particles are estimated at each future time instant to approximate a fault state distribution at that time instant (the first prognosis level). Then, the fault distributions at all the future time instants are compared with the failure threshold $F_f$ by applying the law of total probability to calculate the RUL distribution (the second prognosis level).

This RS-based prognostic approach often involves a large prognostic horizon, especially at the early stage of a fault and when the fault growth is slow. This large prognostic horizon causes two major issues. First, it is computationally expensive and not suitable for applications with limited computational resource. Second, the uncertainty in prognosis is inherent and will accumulate as the prediction horizon increases. When the uncertainty becomes too large, the estimation of the RUL becomes unreliable that cannot be used in decision-making.

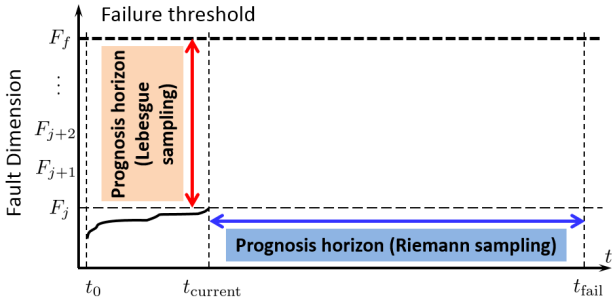



Figure 5. Comparison of prognostic horizon

With LS, the prediction horizon reduces to the number of Lebesgue states from the current Lebesgue state $F_j$ to failure threshold $F_f$. With this idea, each run of the prognostic

algorithm guarantees that the fault has changed and an event has been generated. As a result, a large amount of unnecessary computation can be avoided, which is impossible with RS. It will not only reduce the requirements on computational resources, but also provide an intuitive way to manage uncertainties in prognosis. The comparison of prognostic horizon with RS and LS is illustrated in Figure 5.

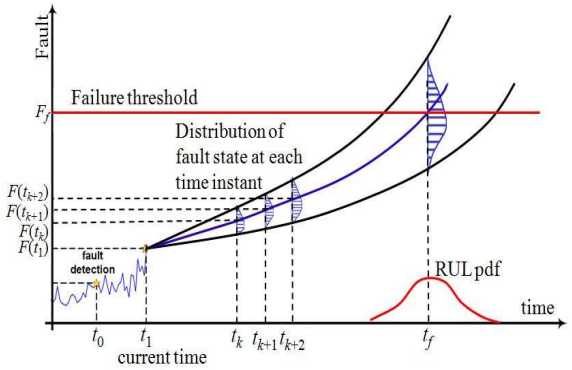
In the context of LS, the prognostic model is given by:

$$t_{k+1} = t_k + g_t(D, \dot{\hat{a}}(t_k)) + \omega_t(t_k) \tag{7}$$

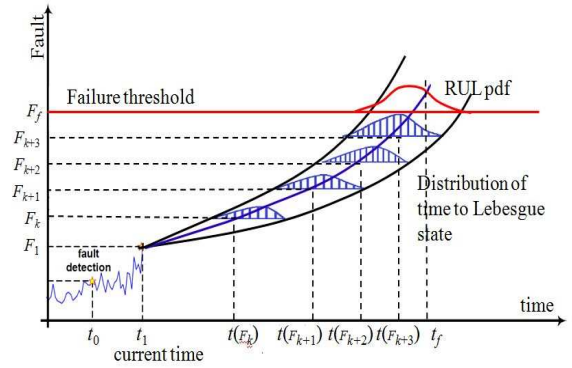
where $D$ is Lebesgue state length and $\omega_t(t_k)$ is a model noise.

With this model, the particles are defined on the time axis instead of the fault dimension axis in RS-based prognosis. To initialize the prognosis, a new set of $N$ particles is defined as $(w_L^{(i)}, t_L^{(i)})$, in which subscript $L$ denotes the Lebesgue state, $w_L^{(i)}$ denotes the particle weight, and $t_L^{(i)}$ denotes particle on the time axis. The initial particles can be equally weighted with $w_L^{(i)} = \frac{1}{N}, \forall i$ or from diagnosis.

Note that the prognosis is carried out with a model given by equation (7). The outcome is the distributions of the operating time for the fault state to reach each Lebesgue state. Therefore, in this LS-based prognosis, the RUL pdf is calculated directly at the Lebesgue state $L = F_f$.




(a) RS-based prognosis




(b) LS-based prognosis

Figure 6. Comparison of RS-based prognosis and LS-based prognosis

The difference between RS-based and LS-based prognosis is illustrated in Figure 6. We assume that a fault is initialized at an unknown time instant $t_0$. The fault is detected at $t_1$ and prognosis is activated from this time instant. For RS-based prognosis in Figure 6(a), the prediction horizon is $[t_1, t_f]$, where $t_f$ is the time stamp when the prediction of all particles pass the failure threshold. With a sampling period of $T$, the prognostic algorithm needs to recursively prediction all particles $(t_f - t_1)/T$ steps and this is the most time-consuming part of prognosis which limits many applications. In other words, the prediction steps are $[t_1, \cdots, t_k, t_{k+1}, t_{k+2}, \cdots]$ on the horizontal time axis. The expectations of the distributions of the operating time to reach these Lebesgue states are $[t_1, \cdots, t(F_k), t(F_{k+1}), \cdots, t_f]$, of which the time intervals could be uneven.

In the Lebesgue sampling-based prognosis, the prediction horizon is $[F_1,\ F_f]$ where $F_f$ is the fault dimension that indicates the failure of the system. With a uniform Lebesgue length of $D$, there will be $(F_f - F_1)/D$ predication steps, and can be denoted as $[F_1, \cdots, F_k, F_{k+1}, \cdots, F_f]$ on the vertical axis. The expectations of the distributions of the operating time for the fault reaching these Lebesgue states are $[t_1, \cdots, t(F_k), t(F_{k+1}), \cdots, t_f]$, of which the time intervals are uneven. In summary, the fundamental difference is that RS-based prognosis calculates fault state distribution at given time instants, while LS-based prognosis calculates time distribution at predefined Lebesgue states.

## 4. EXPERIMENTAL RESULTS

In this section, the proposed LS-FDP scheme with a particle filtering algorithm will be verified in a case study of an epicyclic gear system in which a crack in the planetary carrier plate is developed.

### 4.1. Planetary Gear Box

The main transmission of Blackhawk and Seahawk helicopters employs a five-planet epicyclic gear system, which is a critical component directly related to the availability and safety of the vehicle. The fault is a crack in the planetary carrier plate, as shown in Figure 7.

A timely detection of crack and prediction of failure will not only help the decision-making on mission planning and system reconfiguration, but also improve the reliability and safety of the vehicle. In the experiments, a fault of seeded crack grows with the evolving operation of the gearbox. The gearbox operates over a large number of Ground-Air-Ground (GAG) cycles at different torque levels. An accelerometer is mounted at a fixed position to collect vibration data as crack length grows. In our previous research, the vibration signal processing and feature extraction have been discussed and applied in Riemann sampling based diagnosis and prognosis (Chen, Zhang, Vachtsevanos, & Orchard, 2011; Chen, Zhang,

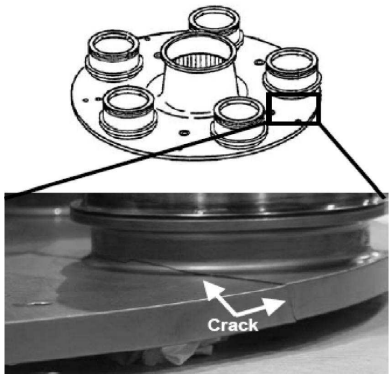



Figure 7. Crack of planetary gear carrier plate.

& Vachtsevanos, 2012; Zhang, Khawaja, et al., 2009; Zhang, Sconyers, et al., 2009; Zhang et al., 2008). In this section, we will use the crack growth data for verification of LS-FDP.

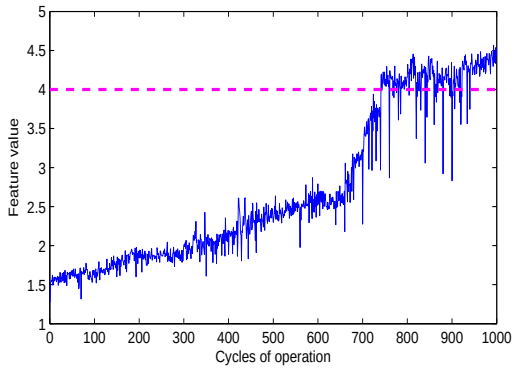



Figure 8. Feature vector for fault growth.

The feature vectors is shown in Figure 8. Since a fault is seeded in the experiment, the data from the first 50 cycles are used as baseline data, which has a mean value of 1.5741, and our objective is to detect the real-time fault growth from this baseline crack length. Note that this feature value will be used as a direct indicator of fault dimension as fault crack length as described in Section 2. For prognosis, the failure threshold is set as 4. The figure shows that the feature value reaches this threshold at around 750th cycle of operation.

#### 4.1.1. RS-based Diagnosis and Prognosis

To implement diagnosis and prognosis, a fault growth model needs to be developed. For Riemann sampling based diagnosis and prognosis, the fault growth model is given by:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + p_1 \cdot a(t_k)^{p_2} + \omega(t_k) \tag{8}$$

where $p_1$ and $p_2$ are parameters and $\omega$ is a model noise.

A particle filtering with 500 particles are implemented and the results of fault diagnosis is shown in Figure 9. The fault is detected at the 183rd cycle, at which the expected value of fault state is 1.94 and the 95% confidence interval is $[1.79,\ 2.08]$.

In this figure, the top subfigure is the feature, given by blue curve, compared with the filtered feature, given by magenta curve. the bottom subfigure shows the comparison of baseline pdf (green one) compared with the real-time estimation pdf (red bars) at the cycle when the fault is detected. In this experimental, 5% false alarm rate is defined and the fault detection threshold is given by the blue vertical line. Note that in this RS-based diagnosis, the diagnostic algorithm needs to execute 183 time, *i.e.*, every time when a new feature becomes available.

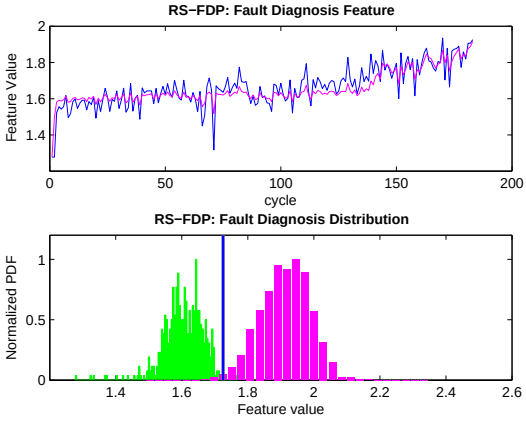



Figure 9. Experimental result of RS-based diagnosis.

As a fault is detected, prognostic algorithm is activated to conduct the long-term predication and estimation of RUL. The initial condition of prognostic algorithm is the fault state at the cycle when the fault is detected. The result of fault growth and RUL estimation is shown in Figure 10. To make the figure clear, only the fault state pdf at the 183rd cycle is plotted and the fault state pdf at other time instants are not shown in this figure. Instead, the expected value, upper and lower bound of 95% confidence interval of the pdf at each time instants are shown in this figure. Note that the prognosis needs to predict all particles from its current value at the cycle 183 to the failure threshold value. In this figure, the prediction horizon is about 700 cycles. To make the real-time implementation of prognosis possible, the number of particles is reduced to 20.

Then, the fault state pdf at each time instant is compared with the failure threshold to obtain the RUL pdf, as shown in the histogram on the horizontal axis. This process uses the law of total probabilities and can be mathematically described as:

$$p_{failure}(t) = \sum_{i=1}^{N} Pr\left(Failure|x_t^{(i)} > F_f\right) w_t^{(i)} \tag{9}$$

where superscript $(i)$ is the index of particles, $p_{failure}(t)$ is the probability of failure at time $t$, $w_t^{(i)} = Pr(x = x^{(i)})$ is

the weight of particles at time $t$, and $x_t$ is the predicted value of a particle at time $t$.

The RUL pdf is shown as the histogram in the figure. With this figure, the predicted expectation of the failure time is at the 588.6 cycle and the RUL life is 405.6 cycle. The 95% confidence bound of the RUL pdf is given as $[443\ \ 767]$. The uncertainty caused by the long prediction horizon is very large. In addition, from feature vector, we can see that the feature value reaches 4 at around 750th cycle. The distance from the predicted expected value to this ground truth value is 161.4 cycle.

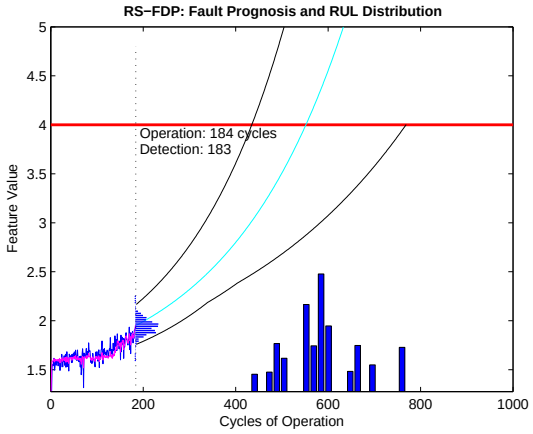



Figure 10. Experimental result of RS-based prognosis.

### 4.1.2. LS-based Diagnosis and Prognosis

For LS-based diagnosis, the feature value range [1.28 4.57] is partitioned into 20 states. The diagnostic algorithm is executed only when the collected feature value changes from one Lebesgue state to another, *i.e.* an event happens. The diagnostic model used in LS-based is given as:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + D \cdot \mathrm{sgn}(\dot{\hat{a}}(t_k)) + \omega_a(t_k) \qquad (10)$$

where $sgn(\cdot)$ is a sign function and $\omega_a$ is the model noise.

The diagnostic results are shown in Figure 11. In the particle filtering algorithm, 500 particles are used. The fault is detected at the 186th cycle. In the upper subfigure of Figure 11, the blue curve is the trajectory of feature values and the magenta curve is the filtered feature from particle filtering. Note that the flat segments mean no event and the diagnostic algorithm does not execute. The lower subfigure shows the fault distribution at the time of detection, where the green distribution is the baseline pdf while the magenta histogram is the real-time fault distribution from diagnosis. The blue vertical line is the threshold of fault detection with 5% false alarm rate. During these 186 cycles, there are 76 events, *i.e.*, the diagnostic algorithm only runs 76 times. The reduction of computational cost is 59.7%, which is a remarkable improvement. At the 186th cycle when the fault is detected, the expected value of fault state is 1.91 and the 95% confidence interval is $[1.69,\ 2.08]$.

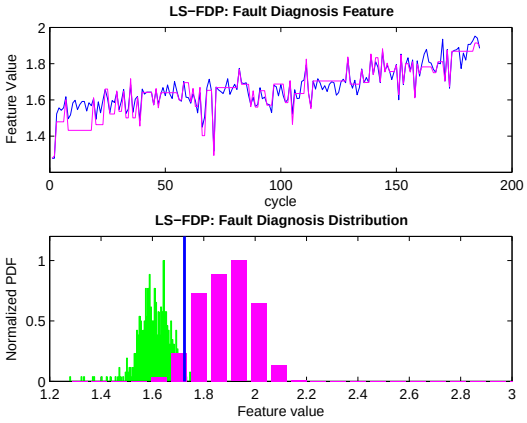



Figure 11. Experimental result of LS-based diagnosis.

Same as the RS-FDP, as the fault is detected, prognostic algorithm becomes activated. Since the prediction horizon is on the vertical axis, the initial condition of prognosis cannot use the estimation result from diagnosis directly. Therefore, we convert the fault state pdf at the time instant of fault detection to a time distribution for fault reaching the current Lebesgue state. This can be done by predicting those particle not yet reach the current Lebesgue state to this state. Then equation (9) is used to obtain the initial time distribution for prognosis. Note that for the prognosis shown in this figure, the prediction horizon is only 15 Lebesgue states, which is very small compare to that in RS-FDP, which is about 700 cycles. Therefore, the LS-FDP prognosis can afford the computation of 500 particles and we do not need to reduce the number of particles.

Since the prognosis is conducted on fault dimension axis, the diagnostic model cannot be used as we described in Section 2. The prognostic model used in LS-based prognosis is given as:

$$t_{k+1} = t_k + D \cdot exp(-\dot{\hat{a}}(t_k))) + \omega_t(t_k) \qquad (11)$$

The prognosis results are shown in Figure 12. To make the figure clear, only the time distribution pdf at a few selected Lebesgue state are plotted. Note that the time distribution pdf at the Lebesgue state defined by the failure threshold gives the RUL estimation pdf. In this figure, the predicted failure time is at the 689.4 cycle and the RUL life is 503.6 cycle. The 95% confidence bound of the RUL pdf is given as $[601\ \ 747.6]$. The uncertainty is much smaller than that of Riemann-sampling based prognosis. When the predicted RUL pdf expected value compared with the ground truth value of 750 cycle, the difference between them is 60.6 cycle.

The advantages of Lebesgue sampling in fault diagnosis and prognosis are obvious from the comparison of above exper-

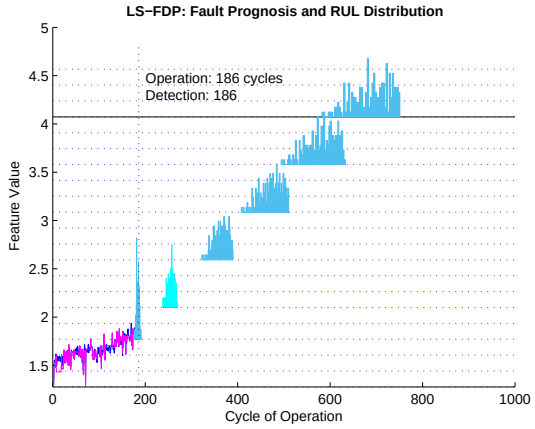



Figure 12. Experimental result of LS-based prognosis.

imental results. For the diagnosis, the two approaches show the comparable performance. In terms of prognosis, the LS-FDP shows better performance in terms of accuracy and precision. First, the introduction of Lebesgue sampling in FDP greatly reduce the computation time and the requirement of computation resources without sacrificing the performance of diagnosis. Since prognosis in Riemann sampling framework usually have a large prediction horizon, it often needs more computation time and resources. This in consequence becomes a main limitation of prognosis for those applications with fault tolerant control and reconfigurable control, where the real-time calculation of RUL is critical. Another important issue with large prediction horizon in Riemann sampling is the significant accumulation of uncertainties in prognosis and the degradation of the performance of prognosis in terms of accuracy and precision. The introduction of Lebesgue sampling in FDP provide a natural solution for real-time implementation, especially on those systems (such as embedded systems) with limited computation capability. The prediction horizon of LS-FDP can be very small comparing to that of RS-FDP, this is very good in managing the uncertainties in prognosis.

## 5. CONCLUSION AND FUTURE WORKS

This paper introduces a novel fault diagnosis and prognosis methodology that aims to: 1) introduce the concept of Lebesgue sampling into FDP and develop a novel FDP approach with an philosophy of "execution only when necessary" or an "as-needed" basis; and 2) enable the FDP on systems with limited computation capabilities, such as the embedded systems, that are widely used in automobiles, distributed diagnosis and prognosis, complex systems and networked systems. The methodology is composed of mathematically rigourous modules including the definition of diagnosis and prognosis in the framework of Lebesgue sampling with particle filtering. Other diagnostic and prognostic algorithms can be applied in this framework similarly. In the LS-FDP, diagnostic model and prognostic model need to be developed separately because fault diagnosis is based on the growth of fault dimension while prognosis is based on the calculation of operation time to reach different Lebesgue states defined as different fault dimensions. Experimental results from RS-FDP and LS-FDP on a planetary gear box with a seeded fault are presented and compared to illustrate the advantages of the proposed solution.

The use of Lebesgue sampling concept in fault diagnosis and prognosis are new in the research community of prognostic and health management. The paper only shows some preliminary results and there are many topics worth further research efforts. Some of the next step research include: 1) In this paper, the Lebesgue states are defined with uniform Lebesgue length. For some applications, the optimal Lebesgue length can be nonuniform and, therefore, the interval between Lebesgue states are not even. 2) Uncertainty management in prognosis is very important and critical. Although LS-FDP in many cases can reduce the prediction horizon and is naturally advantageous in uncertainty management, the theoretical and quantitatively analysis needs to be carried out to provide guidance for FDP algorithm design and implementation. There are many uncertainty management efforts in Riemann sampling based approaches and can be extended to LS-FDP approaches. 3) As we know, modeling is critical to the performance of FDP. The fault growth is a continuous process. For FDP, we discretize the model with Lebesgue sampling and therefore, it is necessary to investigate the accuracy loss caused by Lebesgue sampling. This result will provide a guidance for us on how to optimally choose the Lebesgue states and Exergue length. 4) For many applications, diagnosis and prognosis is not the goal but just the starting point for fault tolerance or system reconfiguration. It is of great interest to integrate the LS-FDP in Riemann sampling-based and Lebesgue sampling-based system reconfigurable control design.

## REFERENCES

Agogino, A., Bonissone, P., Goebel, K., & Vachtsevanos, G. (2001). AI in equipment service. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *15*(4), 265-266.

Anger, C., Schrader, R., & Klingauf, U. (2012). Unscented Kalman filter with gaussian process degradation model for bearing fault prognosis. In *Proceedings of the european conference of the prognostics and health management society.*

Astrom, K., & Bernhardsson, B. (1999). Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of ieee conference on decision and control.*

Balanban, E., & Slonso, J. (2013, Sept). A modeling frame-

work for prognostic decision making and its application to uav mission planning. In *Annual conference of the prognostics and health management society 2013.* New Orleans, LA.

Boskoski, P., & Urevc, A. (2011). Bearing fault detection with application to PHM data challenge. *International Journal of Prognostics and Health Management*, *2*(1), 1-10.

Celaya, J., Saxena, A., & Goebel, K. (2012). Uncertainty representation and interpretation in model-based prognostics algorithms based on Kalman filter estimation. In *Proceedings of the annual conference of the prognostics and health management society.* Minneapolis, MN.

Chen, C., Zhang, B., & Vachtsevanos, G. (2012). Prediction of machine health condition using neuro-fuzzy and Bayesian algorithms. *IEEE Transactions on Instrumentation and Measurement*, *61*(2), 297-306.

Chen, C., Zhang, B., Vachtsevanos, G., & Orchard, M. (2011). Machine condition prediction based on adaptive neuro-fuzzy and high-order particle filtering. *IEEE Transactions on Industrial Electronics*, *58*(9), 4353-4364.

DeCastro, J., Tang, L., & Zhang, B. (2011). A safety verification approach to fault-tolerant aircraft supervisory control. In *Proceedings of aiaa guidance, navigation, and control conference.* Portland, OR.

de Freitas, N. (2002). Rao-Blackwellised particle filtering for fault diagnosis. In *Proceedings of the ieee aerospace conference* (Vol. 4, p. 1767-1772).

Edwards, D., Orchard, M., Tang, L., Goebel, K., & Vachtsevanos, G. (2010). Impact of input uncertainty on failure prognostic algorithms: Extending the remaining useful life of nonlinear systems. In *Proceedings of the annual conference of the prognostics and health management society.* Portland, OR.

Finkelstein, M. (2004). On the exponential formula for reliability. *IEEE Transactions on Reliability*, *53*(2), 265-268.

Goebel, K., Eklund, N., Hu, X., Avasarala, V., & Celaya, J. (2006). Defect classification of highly noisy NDE data using classifier ensembles. In *Smart structures and materials.* San Diego, CA.

Goebel, K., Saha, B., & Saxena, A. (2008, May). A comparison of three data-driven techniques for prognostics. In *Proceedings of the 62nd meeting of the society for machinery failure prevention technology.*

Hess, A., & Wells, S. (2003). Sliding mode control applied to reconfigurable flight control design. *Journal of Guidance, Control,and Dynamics*, *26*(3), 452-462.

Huang, W., & Dietrich, D. (2005). An alternavie degradation reliability modeling approach using maximum likelihood estimation. *IEEE Transactions on Reliability*, *54*(2), 310-317.

Isermann, R. (2005). Model-based fault detection and diagnosis-status and applications. *Annual Reviews in Control*, *29*, 71-85.

Jardine, A., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition based maintenance. *Mechanical Systems and Signal Processing*, *20*, 1483-1510.

Kaminskiy, M. (2005). A simple procedure for Bayesian estimation of the Weibull distribution. *IEEE Transactions on Reliability*, *54*(4), 612-616.

Lawless, J. (2003). *Statistical models and methods for lifetime data*. Hoboken: John Wiley and Sons.

Li, Y., Kurfess, T., & Liang, S. (2000). Stochastic prognostics for rolling element bearings. *Mechanical Systems and Signal Processing*, *14*(5), 747-762.

McCann, R., & Le, A. (2008). Lebesgue sampling with a Kalman filter in wireless sensors for smart appliance networks. In *Proceedings of industry applciaitons society annual meeting.*

Morales-Menendez, R., de Freitas, N., Monterrey, I., Freitas, O. D., & Poole, D. (2002). Real-time monitoring of complex industrial processes with particle filters. In *Nips* (pp. 1433–1440).

Oppenheimer, C. H., & Loparo, K. A. (2002). Physically based diagnosis and prognosis of cracked rotor shafts. In *Aerosense* (pp. 122–132).

Orchard, M., Hevia-Koch, P., Zhang, B., & Tang, L. (2013, Nov). Risk measures for particle-filtering-based state-of-charge prognosis in lithium-ion batteries. *IEEE Transactions on Industrial Electronics*, *60*(11), 5260-5269.

Orchard, M., & Vachtsevanos, G. (2009). A particle filtering approach for online fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, *31*(3/4), 221-246.

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009). Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control*, *31*(3-4), 293-308.

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). On application the prognostic performance metrics. In *Proceedings of international conference on prognostics and health management.*

Schwabacher, M., & Goebel, K. (2007). A survey of artificial intelligence for prognostics. In *Aaai fall symposium* (p. 107-114). LA USA.

Tang, L., Hettler, E., Zhang, B., & DeCastro, J. (2011). A testbed for real-time autonomous vehicle PHM and contingency management applications. In *Proceedings of international conference on prognostics and health management.* Montreal, Canada.

Tang, L., Zhang, B., DeCastro, J., & Hettler, E. (2011). An integrated health and contingency management case study on an autonomous ground robot. In *Proceedings*

*of the 9th ieee international conference on control and automation.*

Tumer, I., & Bajwa, A. (2004). A survey of aircraft engine health monitoring systems. In *Proceedings of the 35th aiaa/asme/ sae/asee joint propulsion conference* (p. 620-625).

Usynin, A., & Hines, J. (2007, November). Uncertainty management in shock models applied to prognostic problems. In *Aaai fall symposium.* Arlington, VA.

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems.* John Wiley and Sons.

Yang, G. (2005). Accelated life test at higher usage rates. *IEEE Transactions on Reliability*, *54*(1), 53-57.

Zhang, B., Khawaja, T., Patrick, R., & Vachtsevanos, G. (2008). Blind deconvolution de-noising for helicopter vibration signals. *IEEE/ASME Transactions on Mechatronics*, *13*(5), 558-565.

Zhang, B., Khawaja, T., Patrick, R., & Vachtsevanos, G. (2010). A novel blind deconvolution de-noise scheme in failure prognosis. *Transactions of the Institute of Measurement and Control*, *32*(1), 3-30.

Zhang, B., Khawaja, T., Patrick, R., Vachtsevanos, G., Orchard, M., & Saxena, A. (2009). Application of blind deconvolution de-noising in failure prognosis. *IEEE Transactions on Instrumentation and Measurement*, *58*(2), 303-310.

Zhang, B., Sconyers, C., Byington, C., Patrick, R., Orchard, M., & Vachtsevanos, G. (2009). A probabilistic fault detection approach: application to bearing fault detection. *IEEE Transactions on Industrial Electronics*, *58*(5), 2011-2018.

Zhang, B., Tang, L., DeCastro, J., & Goebel, K. (2011, Aug). Prognostics-enhanced receding horizon mission planning for field autonomous vehicles. In *Aiaa guidance, navigation, and control conference.* Portland, OR.

Zhong, M., Fang, H., & Ye, H. (2007). Fault diagnosis of networked control system. *Annual Reviews in Control*, *31*(1), 55-68.

## BIOGRAPHIES

**Bin Zhang** received the B.E. and M.E. degrees from the Nanjing University of Science and Technology, Nanjing, China, in 1993 and 1999, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2007. He is currently with the Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA. Before that, he was with R&D, General Motors, Detroit, MI, USA, with Impact Technologies, Rochester, NY, USA, and with the Georgia Institute of Technology, Atlanta, GA, USA. His research interests are prognostics and health management and intelligent systems.

**Xiaofeng Wang** received BS and MS degree in mathematics from East China Normal University in 2000 and 2003, respectively, and obtained his PhD degree in electrical engineering from the University of Notre Dame in 2009. After working as postdoctoral research associate in the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign, he joined the Department of Electrical Engineering as assistant professor at the University of South Carolina, Columbia, in 2012. His research interests include networked control systems, real-time systems, event-based control, robust adaptive control, distributed systems, and optimization.